

1. In this lab, I implemented rollback and recovery features on SimpleDB. I decided to add a CLR record whenever a log record is undone. A CLR record consists of the type, a transaction id, a page data for redo, a record offset of the next undo log record, and a starting offset of the CLR record. The CLR record helps recovery when a crash happens in the middle of an undo, and also helps to skip some log records that have been undone in a new undo.

In rollback(), I find the starting offset of the first log record associated with the aborted transaction. Starting from this offset, I scan the log file and add the starting offset of relevant log records to be undone to a stack until reaching the end of the file.

Specifically, during the scanning, if I read an update record and the record is associated with the transaction, I push the starting offset of the update record onto the stack; if I read a CLR record, I pop an offset off the stack because each CLR record associates with an undone record and the order of CLR records is in reverse order of the recording being undone. I also skip unnecessary bytes to read correct information. After scanning the log file, I then pop offsets off the stack until it's empty. For each offset popped off from the stack, I read the before image in the update record starting at the offset and write it to the file, and then append a CLR record for the undone update record.

In recovery(), I first read the first long of the log file. If it is not -1, which means there is a checkpoint in the log file, I update the tidToFirstLogRecord map based on the records in the last checkpoint record, and keep track of the uncommitted transactions in a set. Then I scan the log file from the last checkpoint (or the beginning of the log file if not exist), and redo all log records until the end of the file. Specifically, if I read an update record, I write the after image of the page to the file; if I read a begin record, I add the new transaction to uncommitted set and update tidToFirstLogRecord map; if I read a commit record, I remove the transaction from uncommitted set; if I read a CLR record, I write the page in the record to the file; otherwise, I don't redo anything. I also skip unnecessary data to read correct information. Now the redo phase is done. To start the undo phase, since I know the set of uncommitted transactions, I factor out the codes for undo from rollback() and put them into a new method undo() which takes a set of uncommitted transactions. So I called undo(uncommitted) to finish the undo phase in recovery.

2. I didn't make any changes outside of LogFile.java besides what I did in part 1.
3. It might be useful to add a unit test where T1 inserts and aborts, T2 inserts and commits, and T3 inserts and aborts.